IN THE CLAIMS

Please amend Claim 21 as indicated.

1. (Previously Presented) A method comprising:

modeling network element commands, events and run-time system data into a data model using a first modeling language, the data model comprising first data;

translating the first data represented in the first modeling language to second data represented in a second modeling language;

storing the second data in the second modeling language in a global data model repository; and

automatically generating code to support an external management interface based on the stored second data in the global repository, the external management interface communicating with the stored second data.

2. (Previously Presented) The method of claim 1 further comprising automatically generating system documentation based on the stored second data.

3. (Previously Presented) The method of claim 2 wherein the generated system documentation corresponds to code generated to support an external management interface.

4. (Previously Presented) The method of claim 1 wherein the first modeling language is structured management information (SMI).

5. (Previously Presented) The method of claim 1 wherein the second modeling language is extensible markup language (XML).

6. (Previously Presented) The method of claim 1 wherein automatically generating code for the external interface includes automatically generating code to implement a command line interface (CLI).

7. (Previously Presented) The method of claim 1 wherein automatically generating code for the external interface includes automatically generating code to implement an Extensible Markup Language interface.

8. (Previously Presented) The method of claim 1 wherein automatically generating code for the external interface includes automatically generating code to implement a Simple Network Management Protocol interface.

9. (Previously Presented) The method of claim 1 wherein automatically generating code for the external interface includes automatically generating code to implement a configuration database.

10. (Previously Presented) The method of claim 1 wherein automatically generating code for the external interface includes automatically generating code to implement Simple Network Management Protocol subagents.

11. (Original) The method of claim 1 wherein automatically generating code for the external interface includes automatically generating code to assist in implementation of an Application Program Interface.

12. (Previously Presented) The method of claim 1 wherein modeling includes modeling the run-time system data from a plurality of sources using at least one of the first modeling language and the second modeling language.

13. (Previously Presented) A system comprising:
a memory comprising a global repository;
a processor electrically coupled to the memory;

a first interface to a plurality of network elements, the first interface being in communication with the global repository; and

a second interface to an external interface, the second interface being in communication with the global repository, wherein the processor is configured to:

model network element commands, events and run-time system data into a data model using a first modeling language, the data model comprising first data;

translate the first data represented in the first modeling language to second data represented in a second modeling language;

store the second data in the second modeling language in the global data model repository; and

automatically generate code to support an external management interface code development based on the stored second data in the global repository, the external management interface communicating with the stored second data.

14. (Previously Presented) The system of claim 13 further configured to automatically generate system documentation based on the stored second data.

15. (Previously Presented) The system of claim 14 wherein the generated system documentation corresponds to a code generated implementation.

16. (Previously Presented) The method of claim 13 wherein the first modeling language is structured management information (SMI).

17. (Previously Presented) The method of claim 13 wherein the second modeling language is extensible markup language (XML).

18. (Previously Presented) The method of claim 13 wherein the global repository is further configured to model operational system data using at least one of the first modeling language and the second modeling language.

19. (Previously Presented) A computer program product, tangibly embodied in a computer storage medium, for executing instructions on a processor, the computer program product being operable to cause a machine to:

model network element commands, events and run-time system data into a data model using a first modeling language, the data model comprising first data;

translate the first data represented in the first modeling language to second data represented in a second modeling language;

store the second data in the second modeling language in a memory comprising a global data model repository; and

automatically generate code to support an external management interface code development based on the stored second data in the global repository, the external management interface communicating with the stored second data.

20. (Previously Presented) The computer program product of claim 19 further configured to automatically generate system documentation based on the stored second data.

21. (Currently Amended) The computer program product of claim ~~19~~ 20 wherein the generated system documentation corresponds to the generated code.

22. (Previously Presented) The computer program product of claim 19 wherein the first modeling language is structured management information (SMI).

23. (Previously Presented) The computer program product of claim 19 wherein the second modeling language is extensible markup language (XML).

24. (Previously Presented) The computer program product of claim 19 wherein the global repository is further configured to model operational system data using at least one of the first modeling language and the second modeling language.

25. (Previously Presented) The computer program product of claim 19 wherein the instructions to cause a machine to automatically generate code for the external interface include instructions to cause a machine to automatically generate code to implement a command line interface (CLI).

26. (Previously Presented) The computer program product of claim 19 wherein the instructions to cause a machine to automatically generate code for the external interface include instructions to cause a machine to automatically generate code to implement a configuration database.

27. (Original) The computer program product of claim 19 wherein the instructions to cause a machine to automatically generate code for the external interface include instructions to cause a machine to automatically generate code to implement SNMP subagents.

28. (Previously Presented) The computer program product of claim 19 wherein the instructions to cause a machine to automatically generate code for the external interface include instructions to cause a machine to automatically generate code to implement an API.

29. (Previously Presented) The computer program product of claim 24 wherein instructions to cause a machine to model operational system data include instructions to cause a machine to model operational system data using at least one of the first modeling language and the second modeling language.